# KEY SERVER FOR SECURING IP TELEPHONY REGISTRATION, CONTROL, AND MAINTENANCE

## FIELD OF THE INVENTION

The present invention is directed generally to securing communications between any client-server or server-server and specifically to securing communications between an application server and a packet-switched telecommunications device.

## BACKGROUND OF THE INVENTION

H.323 is a standard for packet switched multimedia communications. The standard provides for terminals that allow for any combination of multimedia communications, i.e., bi-directional audio and/or video and/or data communications. H.323 terminals include IP telephones or IP softphones. An IP telephone provides communications capability the same as analog or digital telephones provide except that communications are routed via the IP trunk card to the packet switched network rather than via an analog or digital trunk line. An IP softphone is a client-based telephony application for the desktop PC or laptop that has similar functionality as a desktop IP telephone.

The H.323 standard discusses the functionality required for interoperability between H.323 terminals and other ITU (International Telecommunications Union) standard terminals, including H.320 terminals operating on narrowband ISDN, H.321 terminals operating on broadband ISDN, H.322 terminals operating on ISOEthernet and H.310 terminals operating on asynchronous transfer mode (ATM). The H.323 standard provides for four functional units that may reside in a co-located system. Terminals are a

first unit. A second unit, gateways, connects terminals on IP and circuit switched networks (ISOEthernet and ATM are virtual circuit switched networks). A third unit, gatekeepers, provides address resolution, registration of endpoints, admission control, monitor network resources and bandwidth, and maintain data integrity. The fourth unit, multipoint control units (MCU's), is designed to support multiparty conferencing.

A public branch exchange (PBX) is a telephone system that supports an enterprise (college, government office, business, etc.) to switch telephone calls between users within the enterprise. All enterprise users share external telephone lines, i.e., trunk lines, which saves the cost of requiring a line for each user to the telephone company's central office. PBXs have evolved from being proprietary hardware/software systems completely separate from the packet switched network to systems running on servers (now known as telephony servers or application servers), interoperable with other application servers through open standards and operating with the data network. Furthermore, application servers have evolved from strictly routing local and long distance telephone calls over the public switched telephone network (PSTN) to additionally providing the capability to route local and long distance telephone calls over packet switched data networks. These application servers operating on packet switched networks allow the enterprise to reduce costs by maintaining one network instead of two (the data and telephone) and reducing charges from toll calls by routing some calls over the packet switched network.

H.225.0 defines the call signaling (communications) protocol between a H.323 terminal and gatekeeper to perform the functions of registration, admission and status. The Registration, Admission and Status protocol (RAS) facilities the gatekeeper's

management over H.323 endpoints (terminal, gateway, gatekeeper and MCU's) and their request for service. RAS uses an unreliable delivery mechanism, the User Datagram Protocol (UDP), which just makes a best effort to deliver data packets. Hence, there is a need to address the integrity of data within RAS packets and authentication of endpoints. H.225.0 defines the RAS protocol and provides security options for adding security to H.323 endpoints, such as, including authentication of endpoints, data integrity to ensure the packet data is not corrupted while in transit and privacy via data encryption.

Encryption and authentication of any form of communication may use either symmetric or asymmetric cryptography. With symmetric cryptography, a secret key is shared between two or more entities and typically is significantly longer than a PIN, password, or pass-phrase. This key is used to encrypt or sign messages sent by one entity and to decrypt or authenticate the signature of the received messages. In asymmetric cryptography, entities use one key (a private key), to encrypt or sign messages, and a second key (public key) is used to decrypt or authenticate the signature of the message.

H.225.0 RAS protocol supports various methods to allow the gatekeeper and endpoint to exchange messages and define the key exchange method used to initiate the call session. There are of course many ways to exchange keys. The key may be provided out-of-band, such as, manually administering a key in each particular host during manufacture or administration, or keys are sent between two endpoints via a local link. Alternatively, there are several well-known key management protocols, such as Rivest-Shamir-Adleman (RSA), Diffie-Hellman, Oakley, and Internet Security Association and Key Management Protocol (ISAKMP), Encrypted Key Exchange, Derived Unique Key per Transaction (DUKPT) and Kerberos. The first six key management schemes operate

between 2 entities whereas the Kerberos scheme operates between 3 or more entities. Unfortunately, these seven well known key management protocols require either public key cryptography or pre-administration of a strong shared secret key during the registration process.

For example, authentication schemes for networked terminals, such as, Point of Sale Pinpad and Signature terminals use the DUKPT (derived unique key per transaction) method. The DUKPT method of key derivation is currently used by most Pinpad manufacturers and ATM machines. DUKPT is a key generation method where a large number of distributed terminals, such as, Point of Sale Pinpad and Signature terminals communicate with a central controller. A shared master secret key is stored in the controller and in the Point of Sale Pinpad or Signature terminal during the manufacturing or initial administration process. Thereafter, the Point of Sale Pinpad or Signature terminal and controller derive a transaction key for each communication which is calculated from the shared master key and non-secret transaction information, such as the terminal identification, transaction number and customer transaction information. All of this information, except for the customer's personal identification number (PIN), is sent in clear text. DUKPT provides protection such that knowledge of the key used in a previous transaction does not compromise future transactions. Unfortunately, DUKPT requires the preadministration of a secret key for all devices on the network.

Figure 1A is an example of the current H.225.0 RAS registration method. At least one telephone system, the Avaya Communication Manager 10 (ACM) embeds the functions of the gateway and gatekeeper using software although such H.323 functionality could reside on separate servers. Alternatively, the functions of the gateway

and gatekeeper can be performed by hardware or firmware. The ACM is an application server 10 as it requires an endpoint to authenticate itself before it can receive services as would be the case for packet switched devices, 16, 17 or other computer on an enterprise network.

The ACM 10 has a processor 6 and memory 7 which manages the switching of the calls within the ACM 10 and inbound and outbound calls. ACM 10 communicates via a packet switched network to H.323 terminals, such as, IP telephones 17 or IP softphones 16. Similarly, ACM 10 supports any other packet switched device that incorporates the H.323 terminal functionality, such as, a cell phone with IP telephony capability, wireless handset with IP telephony capability, or PDA with IP telephony capability.

ACM 10 also supports legacy analog and digital phones and facsimile machines via analog or digital station cards 2. An analog station card 2, or otherwise known as an analog port board, provides the support for the legacy analog telephones and facsimile machines. While a digital station card 2, or otherwise known as digital port board, provides the support for digital or ISDN desktop telephones.

Telephone calls can be routed over the PSTN using the analog or ISDN trunk boards 1 or over the local LAN using the IP trunk card 9 and LAN card 15. If calls are routed over the Internet 12, the messages are sent via the router 11.

The ACM hard disk-drive 5 stores the call processing and maintenance software; software to allow for administration either via the web or at an administrator's terminal using a graphical user interface or command line interface; configuration and user administered data, such as, extensions and user PINs; and software to perform the

functions of a gatekeeper including RAS registration and software to perform the functions of the gateway.

Hardware and software on resource boards 3 provide resources such as dual tone multi-frequency (DTMF), tone generation, etc. Hardware and software on digital signal processing boards 14 provide resources for voice compression/decompression and packetization/depacketization of voice signals.

Figure 1B shows the ACM 10 general registration scheme to register packet switched devices 16, 17. ACM 10 uses a H.323 challenge/response RAS procedure to authenticate the packet switched device 16, 17. Communication in the RAS channel is mostly in clear text except for the challenge strings that are specifically encrypted. The RAS procedure follows the general sequence:

In step 11, ACM 10 administration of the packet switched device 16, 17 includes administering the extension of each packet switched device 16, 17 and the extension's associated PIN. As part of the registration procedure, the packet switched device 16, 17 connects to an ACM 10 RAS (registration, admission, status) port previously configured on the packet switched device 16, 17. The packet switched device 16, 17 sends an H.323 gatekeeper request message, GRQ, with the packet switched device's 16, 17 extension as part of the message.

In step 12, the ACM 10 searches for the extension in the configuration data and finds the administered PIN for that extension. The ACM 10 uses a random number to build a challenge string of digits that is valid for a short period of time so that the packet switched device 16, 17 can not resend an earlier registration request message (or otherwise known as RRQ). The ACM 10 sends this challenge as part of a gatekeeper

confirm message, GCF, to the packet switched device 16, 17. The ACM 10 performs the same computation the endpoint is performing.

In step 13, when the packet switched device 16, 17 receives the GCF message, the packet switched device encrypts the challenge string with a key derived from the packet switched device's PIN. The packet switched device 16, 17 computes the response to the challenge string using the PIN and sends the computed response as the result to the ACM 10.

In step 14, the ACM 10 verifies the response it received from the packet switched device 16, 17 is the same as the computed response in step 12. If correct, ACM 10 proceeds with the registration of the packet switched device 16, 17.

As is evident, the challenge string of digits is encrypted using a very weak key, the user's PIN, which is at most usually 4-8 digits. An intruder in the network may easily guess the user's PIN. Additionally, the intruder may easily try all possible PIN combinations to find the PIN that produces the same response for a particular challenge. Furthermore, the information sent between the ACM 10 and packet switched devices 16, 17 is in clear text except for the encrypted challenge string in the message. Even after the packet switched device 16, 17 is registered with the ACM 10, the information sent via the H.225.0 call-signaling channel between the packet switched device 16, 17 and ACM 10 is in clear text. Hence, unsecure communications over the communication channel may compromise information such as credit card numbers when sent via a packet switched device 16, 17. Similarly, the weak key does not ensure proprietary information such as file downloads to the packet switched device 16, 17 or user administrative settings, such as, telephone settings remain private and/or authenticated.

## SUMMARY OF THE INVENTION

These and other needs are addressed by the present invention. The present invention is generally directed to a packet-switched communications device that is operable to effect provisioning and/or registration using key information received from an application server. As used herein, a "key" refers to a sequence of symbols used with a cryptographic algorithm for encrypting and decrypting data, "encryption" refers to the act of converting information in a first form into a second form that is a code or cipher, "decryption" refers to the act of converting the information in the second form back into the first form, "provisioning" refers to the act of distributing a unique, (secret) typically symmetric, key to a networked computational component as part of component installation at the location of end use, and "registration" refers to the act of authenticating the packet-switched device prior to admitting the packet-switched device to communication via the network. As will be appreciated, provisioning occurs after the device has been shipped to the customer or end user and occurs on the customer's or end user's premises. Provisioning does not refer to installation of keys in the factory or by an intermediate entity before installation in the customer's network or where the device will be used.

In one embodiment, the invention provides a system and method for securing communications between a packet-switched device, such as an IP telephone, IP softphone, or any other networked communication device, such as a computer acting either as a client or server, registering with an application server (which may be for example a gatekeeper or media server). In the preferred embodiment, a key generating agent, which may or may not be collocated with or resident in the application server,

includes security software to generate and provide at least one unique secret key to the packet-switched communications device and the application server.

A secret key distribution and registration process occurs after the packet-switched communications device powers up or resets. The key generating agent generates a unique secret key and one or more key identifiers associated with the secret key. The secret key generated by the key generating agent includes a large number of digits and is a function including an associated identifier, also known as a "key identifier" or "generator." The associated identifier can be derived from information not uniquely identified with the packet switched device for example, a pseudorandom number, a database of keys and key identifiers, a hash function, etc. Alternatively, the associated identifier may be computed using information uniquely identified with the packet switched communications device, such as a user's PIN, password, name, and/or other personal information and/or the device's address (such as the MAC or IP address), extension, serial number, and the like. The key generating agent provides the secret key and the associated identifier to the packet-switched communications device after establishing a secure communication. The packet-switched communications device provides the associated identifier to the application server. The application server provides the associated identifier to the key generating agent after establishing trusted communications with the key generating agent. The key generating agent provides the secret key to the application server and the application server uses the secret key and associated identifier to authenticate the packet switched device securely instead of depending solely on an extensions's PIN as the key, which is much less secure. After the

registration process is complete, on-going communications between the packet switched device and the application server are secure based on a strong symmetric key.

In another embodiment, the application server can be collocated with the key generating agent and is injected with a seed from which a large enterprise key is derived by the key generating agent. This is affected during administration and component installation. Before registering with the application server, the packet-switched communications device contacts the key generating agent, establishes a private data communication, and receives a strong, unique, device-specific secret key and a key identifier, also referred to as a "generator". The device-specific key is unique and may be derived using the generator and the enterprise master key.

Using this invention, the loss of a secret key provides no advantage to a would-be hacker in determining either the enterprise master key or other device-specific keys. Once the device has a device-specific key and generator, it can securely register with the application server and secure signaling channels as well as receive encrypted data using the device-specific key. Examples of such data include maintenance and download information. This methodology does not require that the communications device have prior knowledge of any keys in the key generating agent.

Additionally, the use of the invention does not require a shared-secret key downloaded into the packet switch device (or client or server) as part of the installation or administration. Instead, a shared secret key is sent to the packet-switched device as part of the provisioning process. Hence, the communication channel between the packet switched device and the application server becomes a secure communications channel

without preprovisioning a strong key and without using public key cryptography. These and other advantages are apparent from the discussion below.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1A is a block diagram of the prior art system.

Fig. 1B is a flow chart of the prior art packet switched device registration scheme.

Fig. 2A is a block diagram according to an embodiment of the present invention.

Fig. 2B is a block diagram of the key generating agent according to an embodiment of the present invention.

Fig. 2C is a block diagram of the web server according to an embodiment of the present invention.

Fig. 2D is a block diagram according to an embodiment of the present invention.

Fig. 2E is a block diagram according to an embodiment of the present invention.

Fig. 2F is a block diagram according to an embodiment of the present invention.

Fig. 3 is a flow chart of the initialization and key distribution process between a packet switched device and key generating agent according to an embodiment of the present invention.

Fig. 4 is a flow chart of the key distribution process between a key generating agent and application server and the registration process between the application server and packet switched device.

## DETAILED DESCRIPTION

Figure 2A is a block diagram of the present invention. Note the same reference numbers are used in different figures to designate the same components. Figure 2A

includes an application server 30, key generating agent 42, which is stored in the key server 34 hard disk-drive 42 independent of the application server, web server 33, administration terminal 35 and packet switched devices, such as, IP telephones 37 or IP softphones 36, network 38, and router 31.

The packet switched device 36, 37 can be any H.323 or SIP (Session Initiation Protocol) compliant terminal or any networked computer acting either as a client or server. H.323 terminals or SIP terminals of which an IP telephone is one example, may provide real-time bidirectional audio, video and data communications. Packet switched devices can also include a cell phone with IP capability, wireless handset with IP capability, or PDA with IP telephony capability.

Additionally, the packet switched device 36, 37 stores telephone configuration (script) files and software files to interface and communicate with the key server 34 during the key distribution process and the application server 30 during the registration process. For example, the packet switched device nonce, Re, is a random number generated by the packet switched device to provide registration sequence replay protection. Re includes 128 bits.

In the preferred embodiment, the packet switched device has the capability to request and receive files via HTTP and hence has software stored in its memory to interface with a web server 33 or any server that utilizes the HTTP protocol. The packet switched device 36, 37 may use other methods to send and receive information or files to the application server 30 and key server 34 other than HTTP, such as, file transfer protocol (FTP) or trivial file transfer protocol (TFTP). Preferably, information and files

are sent in a protocol understood by the packet switched device 36, 37, the application server 30, the key server 34, web server 33 and administration terminal 35.

Application Server 30 is similar to application server 10, the ACM, in Figure 1A. However, application server 30 now includes the security software and protocols to interface and communicate with the key server 34 during the key distribution process and during the novel registration process with the packet switched device 36, 37. For example, application server 30 includes security software to create the application server 30 nonce, Rg, a random number generated to provide registration sequence replay protection during the registration process. Rg includes 128 bits.

Additionally, the application server 30 has a web-based administration process which provides a secure HTTP connection between the application server 30 web browser, the Secure Web Service on the web server 33 and the administration terminal 35 web browser.

The application server 30 in this invention is not limited to a telephony server with H.323 gatekeeper and gateway functionality, such as the ACM. This invention also operates in the SIP environment where the application server is known as a SIP proxy server and the endpoints are SIP endpoints. The application server 30 can be any SIP proxy server that provides similar gatekeeper functions as in the H.323 standards. As of this moment, the SIP standard does not incorporate the concept of the gateway in its standard, that is, connecting terminals on IP and circuit switched networks. Instead, the gateway functionality is incorporated as needed into the product implementing the SIP standard. Furthermore, the invention as described uses H.323 messages but the invention is equally operable for other standards regarding packet switched multimedia

communications, such as, SIP which would require the use of SIP based messages. Finally, an application server 30 is defined as requiring an endpoint to authenticate itself before it can receive services as would be the case for packet switched devices, 36, 37 or any endpoint requesting data from the application server 30, etc. Hence, the application server 30 can be a media server that provides video and phone services, or any type of audio, video and data service etc.

This registration process takes a packet switched device 36, 37 that historically had a weak key due to the small number of digits in its PIN and uses Key server 34 to generate stronger keys and an associated key identifier to improve the registration process by protecting and demonstrating knowledge of the PIN or other identifier associated with the packet switched device 36, 37. Thereby allowing the packet switched device 36, 37 to register with the application server 30 in a secure communications environment. Secure communications is defined as communications where messages are encrypted, integrity checked and at least one endpoint is authenticated. Hence following registration, on-going communications between the packet switched device 36, 37 and the application server 30 is encrypted and integrity checked and the packet switched device is authenticated using a strong shared secret. Compromise of one packet switched device 36, 37 or one symmetric key does not compromise the security of any other packet switch device's 36, 37 key and hence this novel registration process prevents out-of-band attacks.

In this process for securing packet switched device registration, the network 38 has at least one packet switched device 36, 37 and an application server 30 that may or may not be coresident with a key server 34. Communications between the packet

switched device 36, 37 and the key server 34 is secure and the packet switched device 36, 37 authenticates the key server 34. One method to establish secure communications is by the use of public key cryptography, such as certificates, however, use of this novel registration process does not require packet switched device 36, 37 specific digital certificates. Instead, a copy of a server certificate issued by a root certificate authority or a copy of the issuing authority certificate may be downloaded as firmware to the packet switched devices 36, 37 during manufacture or during initial administration.

Communications between the key server 34 and application server 30 are trusted and are achievable by similarly using certificates. Trusted communications is defined as communications where messages are encrypted, integrity checked and endpoints are mutually authenticated. The key server 34 and application server 30 can be initially administered with a server certificate issued by a root authority or a copy of the issuing authority certificate may be downloaded as firmware during manufacture or during initial administration.

The communication between the packet switched device 36, 37 and the application server 30 is secure. The application server 30 does not need to use public key cryptography to authenticate the packet switched device 36, 37 but uses symmetric key cryptography. The application server 30 has knowledge of the packet switched device's 36, 37 PIN or other identifier associated with the packet switched device due to the administration process.

Following the establishment of secure communications, a key server 34 communicates with the packet switched device 36, 37 to provide one unique symmetric (secret) key and associated key identifier for a particular extension. The key server 34

provides the key to the application server 30 for a particular extension after trusted communications are established.

Once the application server 30 has the key from the key server 34, the registration process proceeds. The registration function of the application server 30 authenticates the packet switched device 36, 37 before the packet switched device can communicate with another packet switched device 36, 37. Once registration is complete, a secure communication channel between the packet switched device 36, 37 and the application server 30 is established.

The System Administrator uses the administration terminal 35 to administer both the application server 30, the key server 34, and any other servers as needed, such as, web server 33 or router 31. The administrator uses a secure HTTP connection (e.g., HTTPS) via the administration terminal 35 web browser to administer the various servers. Although, the administration terminal 35 may use other secure means and a graphical user interface (GUI) or command line interface to administer the various servers.

Fig. 2B is a block diagram of the key server 34. The key server 34 includes a CPU 41, LAN card 43, such as Ethernet network card, memory 40 and hard disk-drive 42 to store software, keys and associated identifiers. All of these elements communicate via a data bus. The key server hard disk-drive 42 stores the key generating software that includes the three following security functions:

1. Key generating agent Process: The Key generating agent process supports key generation services for the application server 30 and the Key File Generation process. The Key generating agent uses a cryptographically secure pseudorandom number generator (PRNG) to compute the Enterprise Master Key although other methods exist.

The pseudorandom number generator can be based on secure hash algorithms, such as, SHA-1. To provide information required to generate pseudorandom numbers, the Key generating agent is initialized with a number of bits which is known as a "Seed" that is used to generate the Enterprise Master Key, Km. The Enterprise Master Key, Km is used to generate the Packet Switched Device Master Key, Kg, or otherwise known as the secret key. A person skilled in the art will recognize other alternative equations are available to compute Km for single or multiple key servers 34, for example, Km can be calculated as Km=SHA-1(S) where Km includes 160 bits.

Each Kg is a unique value used to derive other Session Keys and a unique Kg is sent to each packet switched device 36, 37 that requests a secret key from the Key generating agent. Kg, is identified by its key identifier, generator(g), and derived from g and Km. A person skilled in the art will recognize a variety of equations are available to compute Kg, for example, Kg= HMAC-SHA-1-128 (g,Km) where Kg includes 128 bits.

Key generator, g, is a unique key identifier for each Kg, and is incremented after each Kg issues. Generator, g, includes 128 bits. The Key generating agent randomly chooses the starting value of g referred to as $g_0$. To insure uniqueness, $g_0$ includes a field that is unique to the specific Key generating agent, such as its IP address or the MAC address of the server the Key generating agent is resident on, and may include another field unique to the packet switched device 36, 37, such as its extension, EXT, or serial number, MAC address, etc. There is additionally a counter field of sufficient length to avoid wraparound within a long period of time.

The Packet Switched Device Authentication Key, Ka, is used to digitally authenticate messages between the application server 30 and the packet switched device

36, 37. Ka is derived from Kg, a constant and the user's PIN. A person skilled in the art will recognize a variety of equations are available to compute Ka, for example, Ka=HMAC-(SHA-1)-128(Kg, Ca||PIN) where Ca is a constant and Ka includes 128 bits.

The Enterprise Master Key Check Value, CVm, is determined by using the lowest three least significant bytes of the following calculation: CVm=SHA-1(Km). A person skilled in the art will recognize alternative equations to compute CVm. CVm may be exchanged between packet switched devices, 36, 37 to confirm that both devices use a Kg based on the same Km.

The Integrity Check Value, ICV, is an optional element in H.323 registration, authentication and status messages for authentication of message contents. In the preferred embodiment, Ka is used as the key to the hashed message authentication code and one formula that may be used to compute ICV is, for example, ICV=HMAC-(SHA-1)-96. Refer to HMAC as defined in RFC 2104 for other formulas.

The Session Pre-Master Secret, Ks, is used as the transport layer security (TLS) pre-master secret for establishment of a TLS session or as key material to encrypt information exchanged between the application server 30 and the packet switched device 36, 37 within the H.225.0 signaling channel. Ks may be computed as follows: Ks=HMAC-(SHA-1)-128 (Kg, Cb||PIN||Re||Rg), where Cb is a constant and Ks includes 128 bits. A person skilled in the art will recognize other alternative equations to compute Ks.

If the Key generating agent is not located on the key server 34 or on the same server operating the Key File Generation Process then the Key generating agent process should have secure access to the server operating the Key File Generation Process.

2. The Key File Generation Process: This process obtains key information from the Key generating agent and formats it into a file, the Key File, which is in the format required by the packet switched devices 36, 37 and application server 30. Kg is unique for each packet switch device key request, hence Key Files are generated dynamically. The Key File Generation Process must have secure access to the Key generating agent process and to the Secure File Service process that resides on the web server 33. Alternatively, the Secure File Service process resides on the key server 34 or on the application server 30. If secure access is not possible, the Key File could be generated directly by the Key generating agent Process.

3. The Key generating agent web-based administration software: In the preferred embodiment, the Key generating agent is administered through a secure HTTP connection between the administration terminal 35 and the key server 34. The administrator on the administration terminal 35 administers the Key generating agent process by establishing secure HTTP communications with the Secure Web Service residing on the web server 33 and the Key generating agent web-based administration software residing on the key server 34. Note, a web server 33 is not required. Any secure administrative mechanism will work, such as secure shell (SSH).

Fig. 2C is a block diagram of the web server 33. The web server 33 includes a CPU 44, LAN card 46, such as, Ethernet, memory 45 and a hard disk-drive 47. The web server 33 hard disk-drive 47 stores administration pages and the software to implement the following four functions:

1. Secure Web Service Administration software for the application server: This software includes the libraries and associated software to administer the application

server 30 from the administrator's terminal browser 35 via the Secure Web Service, a secure HTTP connection, on the web server 33.

2. Secure Web Service Administration software for the key server: This software includes the libraries and associated software to administer the key server 34 from the administrator's terminal browser 35 via the Secure Web Service, a secure HTTP connection, on the web server 33.

3. Secure File Service: This is an HTTP server process that utilizes the TLS (HTTPS) protocol to encrypt data requested by packet switched devices 36, 37, such as configuration (script) files. Basic Secure File Service operation supports packet switched devices 36, 37 reading (not writing) of data. Basic Secure File Service operation also supports authentication of the file service by the packet switched device 36, 37. Where the packet switched device 36, 37 authenticates the Secure File Service by authenticating the web server 33 certificate chain against a Root Certificate built into the packet switched device's operating software. Note the Basic Secure File Service does not authenticate the packet switched device or the end user. If a file exists, the file is delivered to any endpoint that requests it.

Basic Secure File Service also supports dynamic files, that is, content generated on demand by a process. Basic Secure File Service supports passing of parameters, such as a packet switched device 36, 37 extension, EXT, from the URL of the HTTP GET message requesting the file to the process that generates the file. The packet switched device 36, 37 extension, EXT, is one example of a parameter to identify the packet switched device 36, 37. Other examples may include serial number, user log in id, etc.

Advanced Secure File Service, supports mutual authentication of the application server 30 and the key server 34. Additionally, Advanced Secure File Service supports application server 30 writing of files subject to limitations on file size, number of files, and file naming conventions, etc. The allows the application server and the Advanced Secure File Service of the key server to share information securely whereby the application server retrieves information from the key server necessary to authenticate registration of the packet switched device.

The functions of the web server could be co-resident in the key server 34 or in the application server 30 if desired. For example, the Secure File Service may be an HTTP server process running co-resident on the application server 30 or on the key server 34. The Secure File Service may be the same process the Secure Web Service used for administering the application server 30, the key server 34 and Key generating agent process on the key server 34.

For Fig. 2A, there is no requirement that the application server 30 include the functions of the gatekeeper, gateway or even have web-based administration. The gatekeeper and gateway functions may reside on a separate server. As for the web-based administration for the key server 34 or application server 30, other forms of administration are possible, such as, using a graphical user interface (GUI) or a command line interface using a mouse or voice recognition to implement changes. As for the transfer of files, it does not need to be via HTTP but can be via any other protocol that can be secured and is understood by the application server 30 and packet switched devices 36, 37.

For a small organization or enterprise, the functions of the web server 33 may be co-resident on the application server 30 or key server 34. Furthermore in small organizations, the functions of the key server 34 may be co-resident with the application server 30 depending on the processing power of the application server and system usage. Alternatively depending on the system usage, the key server 34 may remain on a separate server from the application server. Figures 2D, 2E and 2F are block diagrams for a second, third and fourth embodiment of the invention denoting the case where the functions of the web server 33 are co-resident on the application server 30 or key server 34.

For larger organizations (i.e., enterprises) that have multiple application servers 30 in a building, the needs of the organization may be met with a single key server 34 as shown in Fig. 2D. For high availability purposes or perhaps to improve system performance, multiple key servers 34 serve the key distribution needs of the multiple application servers 30 and packet switched devices 36, 37 as shown in Fig. 2E. One issue that arises in this case, is the need to administer the Key generating agent process operating on each key server 34. This includes entering the same Seed during the Key generating agent administration process on each key server 34 so that each Key generating agent process produces the same Enterprise Master Key, Km.

Finally there may be a need in a large organization that has several remote offices 55, to have an application server 49 with co-resident key server functionality in each of the remote offices 55 as shown in Fig. 2F. Where each of the remote offices 55 communicate via the router 31 and Internet. In this case, the key server functionality is co-resident with the application server 49 because the small number of users in each

remote office 55. For high availability uses, if one application server 49 with co-resident key server functionality fails in a first remote office 55, it may be desirable to have a remote application server 49 or remote application server 30 and key server 34 in a second office 55 take over until operations are back to normal in the first remote office 55. Again, the need to administer the Key generating agent operating on application server 49 includes entering the same Seed during the Key generating agent administration process on each remote key server 34 so that each Key generating agent in the enterprise produces the same Enterprise Master Key, Km.

Fig. 3 is a flow chart of the initialization and key distribution process between a packet switched device 36, 37 and key server 34. Note for flow charts in Fig. 3 and 4, the Secure File Service is co-resident on the key server 34 rather than residing on the web server 33.

There are several system independent procedures listed in step 50. When the packet switched device 36, 37 powers up or resets, it broadcasts a DHCP DISCOVER message. A DHCP server (not shown) issues the packet switched device 36, 37 an IP address and the IP address of the server operating the Secure File Service in addition to other information. Alternatively, the packet switched device 36,37 may use a static IP address.

In step 51, the packet switched device 36, 37 attempts to establish a secure connection using a TLS Handshake protocol process using HTTP over TLS (HTTPS) as follows:

1. The packet switched device 36, 37 and Secure File Service on the key server 34 establish a logical connection using the HTTPS port 443 or any other designated port.

2. The packet switched device 36, 37 and Secure File Service exchange Hello messages to agree on security capabilities including cryptographic algorithms the client can support (such as RSA), the session ID ("SID"), TLS protocol version and 32 byte random numbers to seed the cryptographic calculation of the symmetric key (shared secret key).

3. The Secure File Service sends a Certificate message including the key server's (the key server 34 is operating the Secure File Service) public key certificate and the certificate authority's root certificate. The packet switched device 36, 37 can authenticate the server by authenticating the certificate chain, i.e., the public key and certificate authority's root certificate. Packet switched device 36, 37 has compiled in its software a copy of the Root Certificate Authority certificate which was downloaded and compiled during manufacture or at a later time, such as, during initial administration. The Secure File Service requests a Client Key Exchange message from the packet switched device 36, 37.

4. If the certificate chain is verified, the packet switched device 36, 37 sends the Client Key Exchange message. The Client Key Exchange message uses the public key contained in the server's public key certificate to encrypt the session key information (shared secret key) included in the message. The packet switched device 36,37 sends a ChangeCipherSpec message to activate the negotiated options for all messages the packet switched device 36, 37 will send. The packet switched device 36, 37 also sends a Finished message to let the Secure File Service verify the activated options.

5. The Secure File Service sends a ChangeCipher Spec message to activate the negotiated options for all messages the Secure File Service will send. The Secure File

Service sends a Finished message to the packet switched device 36, 37 to notify the packet switched device 36, 37 it should verify the activated options.

In step 52 when the Secure File Service is authenticated, the packet switched device 36, 37 uses the SID to establish a secure communications channel using HTTP over TLS (HTTPS).

In step 53, the packet switch device 36, 37 requests the Packet Switched Master Key, Kg, and generator, g, from the Secure File Service. This is the beginning of the key distribution process between the packet switched device 36, 37 and the key server 34. The key distribution process is required if the packet switch device 36, 37 rebooted, manually logged out, changed extensions, or is brand new. The key distribution process is also required if the application server 30 determined that any portion of a previous registration request message is incorrect or too old and hence did not authorize the packet switched device's 36, 37 registration request.

If the key distribution process is required, the packet switched device prompts the user for an extension (EXT) and PIN. The EXT and PIN are stored for the moment in the packet switched device's 36, 37 volatile memory. Once the user replies, the packet switched device sends to the Secure File Service an HTTP GET for the URL formed by "https://" including the IP address of the Secure File Service, the Key file path name and file name "kkg.txt", and the packet switched device's extension. The extension or alternative unique identifier associated with the packet switched device 36, 37may be sent to the key generating agent to derive g for this particular extension. In an alternative embodiment, the key generating agent may send to the packet switched device 36, 37 a g derived from information that is independent of the packet switched device. In step 54

upon receipt of the request, the Secure File Service passes the request to the Key File Generation process. The Key File Generation process obtains a unique shared symmetric key, Kg, and key identifier, g from the Key generating agent process. In an alternative embodiment, g may be computed by the packet switched device 36, 37 and sent as part of the request for the Packet Switched Master Key, Kg.

The Key generating agent process also computes a Master Key Check Value (CVm) for identification of the Enterprise Master Key, Km. The Key File generation process returns to the Secure File Service a Key File with the information in a format appropriate for the packet switched device 36, 37. The Secure File Service sends the Key File to the packet switched device 36, 37. The packet switched device 36, 37 stores Kg, g and CVm in non-volatile memory.

The packet switched device 36, 37 closes the secure connection, computes the Packet Switched Device Authentication Key, Ka, which is used to digitally sign messages between the packet switched device, 36, 37 and the application server 30. The packet switched device 36, 37 stores Ka in non-volatile memory.

Fig. 4 is a flow diagram of the key distribution process between key server 34 and application server 30 and the registration process between the application server 30 and packet switched device 36, 37. Note, if a packet switched device has a Packet Switched Device Authentication Key, Ka, generator, g, extension (EXT) and PIN in non-volatile memory, then it does not attempt the key distribution process first but instead tries to register first with the application server 30. If the registration process fails, for example due to an obsolete g and CVm because the Enterprise Master Key, Km, changed, the packet switched device should begin the key distribution process as shown in Fig. 3.

In step 60, the packet switched device 36, 37 connects to application server 30 RAS (registration, admission, status) port previously configured on the packet switched device 36, 37. The packet switched device 36, 37 sends the application server 30 a gatekeeper request message, GRQ, to register with the application server 30. The GRQ message includes generator, g, packet switched device 36, 37 random number, Re, the packet switched device 36, 37 extension, EXT, Master Key Check Value, CVm, and an Integrity Check Value, ICV, based on the Packet Switched Device Authentication Key, Ka. In an alternative embodiment, ICV may be based on Kg. Note that this authentication mechanism does not require the transmission of the Packet Switched Device Authentication Key or the PIN or combination thereof. Furthermore, this mechanism does not require the receipt of a challenge message from the Application Server before sending the GRQ message.

The Enterprise Master Key Check Value, CVm, provides the ability for the application server 30 to determine if the packet switched device is using a Packet Switched Device Master Key, Kg, that is derived from the correct Enterprise Master Key, Km. Furthermore, by computing an ICV using Ka, the packet switched device 36, 37 proves it knows the PIN for the extension (EXT) without having to send the PIN in the GRQ message. Note the GRQ message is sent in clear text but the messages are authenticated. Hence, a third party sniffing the clear text messages can not impersonate a packet switched device 36, 37. A third party can replay the message but cannot compute a new ICV.

In step 61, the key distribution process between the key server 34 and application server 30 begins. Upon receipt of the GRQ message from the packet switched device 36,

37, the application server 30 requests authentication from the Advanced Secure File Service on the key server 34 to establish a trusted communication channel and receive a Packet Switched Device Master Key, Kg associated with the generator, g, and extension (EXT) received in the GRQ message.

Establishing trusted communications between the application server 30 and the Advanced Secure File Service of the key server 34 allows for mutual authentication. The trusted communications process can be performed once as long as the session is maintained and reused. This trusted communications between the application server 30 and the key server 34 is via the HTTP over TLS (HTTPS) process as follows:

1. The application server 30 and Advanced Secure File Service operating on the key server 34 establish a logical connection using the HTTPS port 443 or any other designated port.

2. The application server 30 and Advanced Secure File Service exchange Hello messages to agree on security capabilities and parameters including cryptographic algorithms the client can support (such as RSA), the session ID ("SID"), protocol version and 32 byte random number to seed the cryptographic calculation of the symmetric key (shared secret key).

3. The Advanced Secure File Service sends a public key certificate in a Certificate message to the application server 30. The public key certificate includes the File Service's public key. The Advance Secure File Service sends a CertificateRequest message to notify the application server 30 the Advance Secure File Service wants to authenticate the application server 30. The CertificateRequest message includes a list of certificate types the key server 34 will accept and a list of certificate authorities the key

server 34 will accept. The Advanced Secure File Service requests a Certificate from the application server 30 to prove knowledge of a related private key.

4. The application server 30 sends a Certificate message including the application server's public key certificate and the certificate authority's root certificate. The Advanced Secure File Service on the key server 34 can authenticate the application server 30 by authenticating the certificate chain, i.e., the public key and certificate authority's root certificate. The key server 34 and application server 30 have compiled in their software a copy of the Root Certificate Authority certificate, which was downloaded and compiled during initial administration. The Advanced Secure File Service requests a Client Key Exchange message from the application server 30. The application server 30 sends the Client Key Exchange message which includes the public key contained in the application server's 30 public key certificate to encrypt the session key information (shared secret key) included in the message.

5. The application server 30 also sends a CertificateVerify message, which authenticates the key information via a keyed cryptographic hash of the information exchanged in the handshake messages. This allows the Advanced Secure File Service to prove the application server 30 has the appropriate private key.

6. The application server 30 sends a ChangeCipherSpec message to activate the negotiated options for messages it will send to the Advanced Secure File Service. The application server 30 also sends a Finished message to notify the Advance Secure File Service to check the activated options.

7. The Advance Secure File Service sends a ChangeCipherSpec message to activate the negotiated options for messages it will send to the application server 30. The

Advanced Secure File Service sends a Finished message to notify the application server 30 to check the activated options.

In step 62, if authentication is successful, the application server 30 uses the SID to establish the trusted communications. Once the trusted communication is established, the Key generating agent provides Kg associated with the generator, g, and extension (EXT) received by the application server 30 in the GRQ message in the following manner:

1. Upon receipt of the request, the Advanced Secure File Service passes the request to the Key File Generation process. The Key generating agent also computes a Master Key Check Value (CVm) to assure usage of the same Km. The Key File Generation process provides the unique symmetric key, Kg, for the particular identifier, g, sent by the application server 30. The Key File generation process returns to the Advanced Secure File Service a Key File with the information in a format appropriate for the application server 30. The Advanced Secure File Service sends the Key File to the application server 30.

2. The application server 30 optionally closes the trusted communication channel with the Advanced Secure File Service of the key server 34.

Step 63 continues the registration process between the packet switched device 36, 37 and the application server 30. The application server 30 verifies the contents of the received GRQ message now that it has the symmetric (secret key) key, Kg, from the Advanced Secure File Service. Once the GRQ message is authenticated, the application server 30 sends, via the RAS port, a gatekeeper confirm message, GCF, to the packet switched device 36, 37. The GCF message includes application server 30 nonce, Rg, and

echoes the packet switched device nonce, Re, that the application server 30 received. The message also includes the integrity check value, ICV, computed based on Ka.

Note, the application server 30 nonce, Rg, is used to protect against replay attacks. Furthermore, echoing the packet switched device nonce, Re, proves to the packet switched device 36, 37 that the message is not a replay of an earlier application server 30 message. Signing the message using Ka, proves the application server 30 knows the packet switched device's PIN and the Enterprise Master Key, Km, from which Ka is derived.

In step 64, the packet switched device 36, 37 receives the GCF message and authenticates the message contents. If the contents are correct, the packet switched device 36, 37 sends a request registration message, RRQ to the application server 30. The RRQ includes the application server 30 nonce, Rg, the packet switched device nonce, Re and an ICV computed based on Ka. Note, the use of Rg confirms to the application server 30 that the message is not a replay of an earlier packet switched device 36, 37 message.

In step 65, the application server 30 authenticates the contents of the RRQ message and if valid, sends a registration confirm message, RCF. The RCF includes a session ID (SID), echoed Re, Rg and ICV computed based on Ka. Note that this authentication mechanism does not require the transmission of the Packet Switched Device Authentication Key or the PIN or combination thereof. Furthermore, this mechanism does not require the receipt of a challenge message from the Application Server before sending the GRQ message. Additionally, the strength of this authentication mechanism is

based on the use of the strong symmetric key associated with the key identifier and as such is immune to an attack based on trying all possible symmetric keys.

In step 66, the packet switched device 36, 37 computes the Session Pre-Master Secret Key, Ks, from which it calculates the TLS session master key as described in RFC 2246. The TLS session master key is used to compute cryptographic data for the TLS session.

In step 67, the application server 30 registers the packet switched device 36, 37. Once the packet switched device 36, 37 is registered, the packet switched device 36, 37 establishes a TLS Signaling Channel connection using the Session ID (SID) received as part of the RCF message and the Session Pre-Master Secret, Ks. If registration is successful, the EXT, PIN and key information is stored in non-volatile memory (if different than what is already stored in non-volatile memory) of the packet switched device 36, 37 and registration is complete.

If registration fails due to an invalid EXT, a GRQ is sent to an alternate application server 30. However, if registration fails due to an invalid PIN, the packet switched device must request the end user to login again, begin the key distribution process and registration process again. If registration fails due to missing or invalid key information, the packet switched device must begin the key distribution process and registration process again.

If the application server 30 determines a replay is occurring, it should not respond to the packet switched device 36, 37. For example, if CVm is correct and the application server 30 knows the EXT and its PIN but the ICV value is incorrect, the application server 30 should not respond to the GRQ or any messages that follow. An incorrect ICV

implies the packet was damaged or the packet switched device 36, 37 does not know its PIN or Ka.

If the application server 30 is providing redirection, that is, telling the packet switched device 36, 37 it is not the correct application server for this EXT, and the supplied Master Key Check Value, CVm, is current, then the application server 30 may return to the packet switched device 36, 37 a gatekeeper reject message, GRJ. The GRJ includes a list of alternate application servers 30 to contact, the packet switched device nonce, Re, and error information. The packet switched device 36, 37 receiving a GRJ should delay responding to the message for a reasonable amount of time because a valid GCF message may be returned and in order to avoid a denial of service attack based on fake GRJs.

If multiple unsuccessful login attempts are made from the same IP address, the application server 30 must ignore them to prevent an active PIN guessing attack against an extension.

A number of variations and modifications of the invention can be used. It would be possible to provide for some features of the invention without providing others.

For example in one alternative embodiment, the logic of the present invention is implemented as software, hardware (e.g., logic circuit), or as a combination thereof.

In another alternative embodiment, the key generating agent process is collocated with the application server.

The present invention, in various embodiments, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various embodiments, subcombinations, and subsets thereof. Those of skill in

the art will understand how to make and use the present invention after understanding the present disclosure. The present invention, in various embodiments, includes providing devices and processes in the absence of items not depicted and/or described herein or in various embodiments hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and\or reducing cost of implementation.

The foregoing discussion of the invention has been presented for purposes of illustration and description. The foregoing is not intended to limit the invention to the form or forms disclosed herein. Although the description of the invention has included description of one or more embodiments and certain variations and modifications, other variations and modifications are within the scope of the invention, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative embodiments to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.